

NoSQL in der Cloud - Why?

Andreas Hartmann
adesso AG
Dortmund

Schlüsselworte

NoSQL, Dokumentorientiert, Key Value, Spaltenorientiert, Graph, Business Case, Public Cloud, Private Cloud

Einleitung

Wirft man einen kritischen Blick auf die Buzzwords Cloudcomputing und NoSQL stellt sich automatisch die Frage was hat das miteinander zu tun und welche Vorteile bringt es tatsächlich für das Tagesgeschäft. Im Rahmen dieser Session werden zunächst Cloudcomputing und NoSQL separat betrachtet und dann auf den Vorteil eines kombinatorischen Ansatzes einzugehen. Abschließend werden verschiedene Einsatzszenarien, wie z.B. hybride Persistenzlösungen, aus Architektursicht vorgestellt.

Cloud Computing

Die Motivation für Cloud Computing lässt sich recht schnell herleiten. So fängt es im Projektgeschäft meist damit an, dass im Rahmen des initialen Aufsetzens von Test- und Produktionsserver entsprechende Kapazitätsplanungen vorgenommen werden müssen, wobei das System gerade erst entwickelt wird und das konkrete Benutzerverhalten nicht unbedingt zuverlässig vorhersagbar ist. Im schlimmsten Fall hat ein solches Szenario die Präzision wie der Blick in eine Glaskugel zum Vorhersagen der Lottozahlen. Konsequenterweise wird dies dann während der ersten Systemeinrichtung mit der Beantragung von diversen Rechten auf den Systemen, um die notwendigen Konfigurationen vornehmen zu können. Ist das System dann endlich ausgerollt, kann man durchaus auch mal negativ von dem eigenen Erfolg der Anwendung überrascht werden, indem beispielsweise die Rechenleistung oder der Plattenplatz nicht mehr ausreicht. Das ganze wird nicht selten von diversen Meetings mit der IT-Administration flankiert und mit einer nicht geringen Anzahl von Tickets abgerundet bis die entsprechenden Systeme zur Verfügung stehen und stabil laufen. Dieses Vorgehen ist zwar altbewährt, aber sicherlich optimierungsfähig. Hier bietet Cloud Computing mit seinem on-Demand-Konzept eine vielversprechende Alternative.



NoSQL

NoSQL rangiert, genauso wie Cloud Computing, bei den Hype-Themen momentan ganz oben auf der Skala. Deshalb stellt sich auch hier natürlich sofort die Frage, was bringt mir der Einsatz von NoSQL-Lösungen für meinen Business Case? Betrachtet man die Anwendungslandschaft mal etwas näher, stellt sich heraus, dass diese sich weg von der reinen Verwaltung homogener Daten wandelt hin zu vernetzten Daten, deren

Struktur sich im Laufe der Zeit ebenfalls verändert. Durch die Vernetzung von einzelnen, auf den ersten Blick für den ordinären Business Case nicht so wertvoll erscheinenden Daten, kann man durch die entsprechende Verknüpfung dieser Daten durchaus ganz neue Erkenntnisse gewinnen oder gar neue Produkte generieren. Diese Domäne war bis dato den Business-Intelligence-Speziallösungen einiger weniger Hersteller vorbehalten. Dieses Einsatzszenario kann sehr schnell zu exponentiell wachsenden Datenbestand führen. Hierbei stößt man dann bei herkömmlichen RDBMS sehr schnell an die Grenzen des technisch machbaren, da diese nur bedingt horizontal skalierbar sind. Des

Weiteren sind die RDBMS konzeptionell nicht auf sich häufig ändernde Datenstrukturen und die Vernetzung von Daten ausgelegt.

An dieser Stelle spielen die verschiedenen NoSQL-Lösungen ihre jeweiligen Stärken aus und sind eine ernstzunehmende Alternative zu herkömmlichen RDBMS. Wo Licht ist, ist natürlich auch Schatten. So muss man sich nach Brewer immer für zwei der folgenden drei Eigenschaften Consistency, Availability und Partition tolerance entscheiden. Eine sehr schöne visuelle Einordnung der verschiedenen NoSQL Varianten ist hierzu im Blog von Nathan Hurst zu finden, siehe nachfolgenden Abbildung 1.

Visual Guide to NoSQL Systems

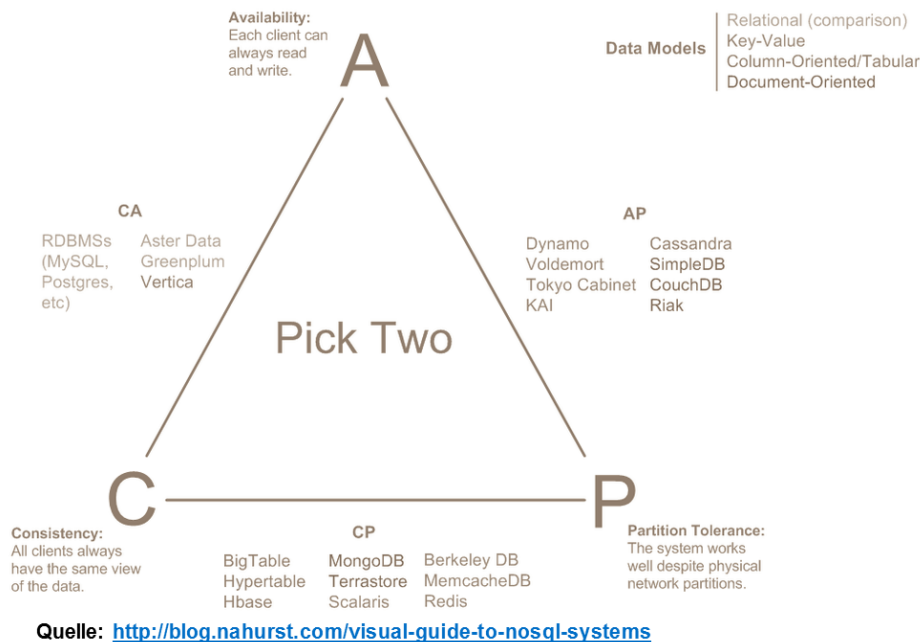


Abb. 1: Visual Guide to NoSQL Systems

Des Weiteren funktionieren die meisten NoSQL-Lösungen nach dem BASE-Prinzip und unterstützen meist keine verteilten Transaktionen. Unter BASE ist kurz zusammengefasst folgendes zu verstehen:

- **Basically Available**
Das System garantiert die Verfügbarkeit, im Rahmen des CAP Theorems
- **Soft-State**
Der Systemzustand kann sich im Laufe der Zeit auch ohne unmittelbare Eingaben ändern
- **Eventual Consistency**
Das System wird im Laufe der Zeit konsistent, insofern keine weiteren Eingaben erfolgen
Allerdings bieten die meisten NoSQL-Lösungen durch ihre horizontale Skalierbarkeit einen alternativlosen Lösungsansatz wenn es um BigData geht. Die volle Stärke wird hierbei dann durch den kombinatorischen Ansatz von Cloud Computing und NoSQL ausgespielt.

Hybride Persistenzlösungen

Betrachtet man beispielhaft die Seiten eines fiktiven Onlinehändlers so sind dort drei Bereiche mit unterschiedlichen Anforderungen hinsichtlich der Konsistenz und Aktualität der Daten denkbar (siehe Abbildung 2). Der erste Bereich bezieht sich auf die bereits ausgewählten Produkte. Diese Daten sind als Business Critical einzustufen und sollten hinsichtlich der Read/Write Operationen transaktional sein um hier einen konsistenten Stand zu garantieren.

Der Zweite Bereich beinhaltet die verfügbare Anzahl zu einem Produkt. Auf diese Daten werden mehr Lese- als Schreib-Operationen ausgeführt. Kurzfristige Inkonsistenzen sind hier hinnehmbar. Der dritte Bereich beinhaltet die Produktbeschreibung. Auf diese Daten wird fast nur lesend zugegriffen und es bestehen keine hohen Ansprüche hinsichtlich der Konsistenz da die Daten selten geändert werden.

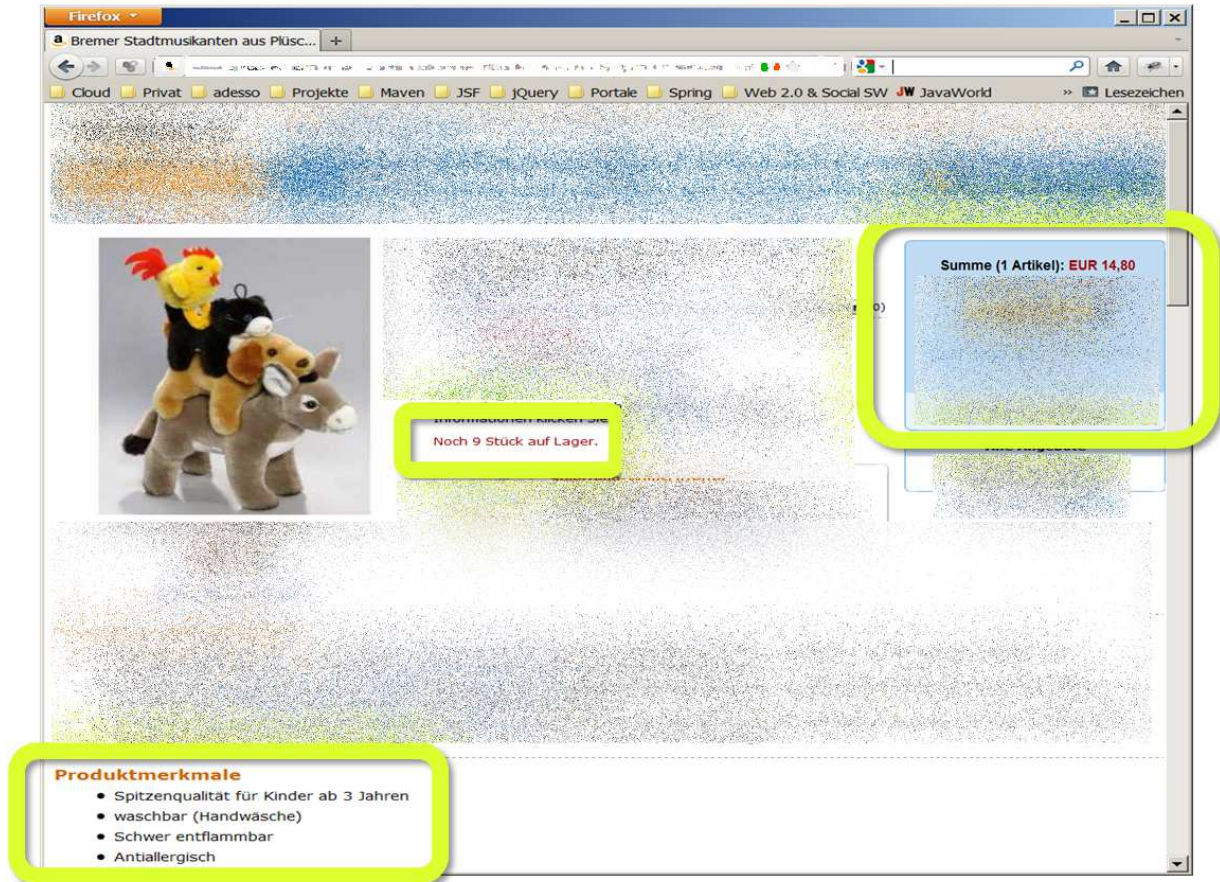
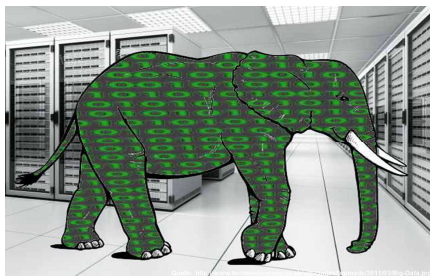


Abb. 2: Seitenstruktur eines Onlineshops

Fazit



Wie im täglichen Leben auch, ist reines Schwarz-Weiß Denken nicht wirklich der eleganteste Weg. So sollte bei dem jeweiligen Business Case zunächst hinterfragt werden, ob ein zusätzlicher Business Value durch den Einsatz von Cloud Computing und NoSQL generiert werden kann und inwiefern eine hybride Persistenz-Lösung, bestehend aus einem herkömmlichen RDBMS und eine NoSQL- Datenbank, nicht der bessere Lösungsweg ist.

Kontaktadresse:

Andreas Hartmann
adesso AG
Stockholmer Allee 24
D-44269 Dortmund

Telefon: +49 (0) 178 2808012
E-Mail hartmann@adesso.de
Internet: www.adesso.de